Northumbria Research Link

Citation: Ogah, Ogah E., Binns, Richard and Sexton, Graham (2008) The HSS/SNiC : a conceptual framework for collapsing security down to the physical layer. In: Proceedings of the 9th Annual Postgraduate Symposium on the convergence of Telecommunications. PGNET, Liverpool, pp. 285-288. ISBN 978-1-902560-19-9

Published by: PGNET

URL:

This version was downloaded from Northumbria Research Link: http://nrl.northumbria.ac.uk/id/eprint/1484/

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: http://nrl.northumbria.ac.uk/policies.html

This document may differ from the final, published version of the research and has been made available online in accordance with publisher policies. To read and/or cite from the published version of the research, please visit the publisher's website (a subscription may be required.)

Northumbria University NEWCASTLE

University Library

The HSS/SNiC: A Conceptual Framework for Collapsing Security down to the Physical Layer

U.E. Ogah*, R. J. Binns*, G. Sexton*
*School of Computing Engineering and Information Sciences, Northumbria University
Newcastle-upon-Tyne, United Kingdom

email:udu.ogah@unn.ac.uk, richard.binns@unn.ac.uk, graham.sexton@unn.ac.uk

***Abstract*— This work details the concept of a novel network security model called the Super NIC (SNIC) and a Hybrid Super Switch (HSS). The design will ultimately incorporate deep packet inspection (DPI), intrusion detection and prevention (IDS/IPS) functions, as well as network access control technologies therefore making all end-point network devices inherently secure. The SNIC and HSS functions are modelled using a transparent GNU/Linux Bridge with the Netfilter framework.**

I. Introduction

In today's network security outfit, Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS) and Stateful Packet Inspection (SPI) firewalls are the current trends that play a major role in securing networks from abuse and malicious attacks [1].

However, these systems are incapable of, for instance, detecting and responding to day-zero vulnerabilities in a web or mail server. Hence, the most lethal and successful compromises on networks have exploited vulnerabilities that stateful firewalls and IDSs cannot normally mitigate. This owes to the fact that these systems are ignorant of the idiosyncrasies of the various application-level protocols. There arises the need for infrastructure that better understand various application protocols, and which can closely examine the packets at wire-speed to detect and dynamically isolate any anomalies/suspicions in the network traffic.

In this paper, the concept and design goals of a novel security architecture is presented as part of ongoing research. The Hybrid Super Switch (HSS) and Super Network Interface Card (SNIC), mitigate issues such as spoofing of MAC addresses at the data link layer of the OSI model and ultimately move up to content-based filtering in layer-7 of the OSI network model. This novel model will reduce processing overhead by distributing the processing work between the HSS and the SNIC made feasible by previous research [2]–[6].

A long-term aim in the research is to incorporate the SNIC and HSS functions into ASICs and network processors to achieve processing efficiency at high network throughput. The authors envisage that network interface cards and switches will have these technologies as basic. The solution also aims to be non-proprietary in nature.

The paper is structured as follows: A background on the typical network security deployments is given including an overview of past and present network security solutions. The NAC concept is introduced with a critique of vendor specific implementations. The ensuing sections then deal with tests conducted on a prototype/generic corporate network detailing how the proposed security model can mitigate a day-zero exploit and concludes by detailing the thrust of future research efforts.

II. Related Work

Recent advances in endpoint security has resulted in commercial vendors [7]–[9] introducing their own implementations to proffer solutions and establish their market share. Some of these network admission and control (NAC) implementations require software agents to be deployed in endpoint machines making deployment clumsy and obtrusive. Other equipment vendors [10], [11] require additional standalone server infrastructure. This may be as a result of vendors' decision to build upon their existing infrastructure base hence possibly inheriting flawed security implementations. Intel's vPro technology [12] while very promising, is primarily meant for out-of-band management of end-users' computers and also requires a very high specification machine as a minimum requirement. Microsoft's network access protection (NAP) [9] while appearing to be a NAC solution, is mainly used to enforce endpoint device compliance with current patches and virus updates prior to being granted network access. The HSS/SNiC framework proposed integrates security into the basic building blocks of PC networks—the switch (wired or wireless) and the endpoint network interface card. Past and current research efforts [13]–[16] also indicate this paradigm shift to deploying security into end-user devices though practical real-world deployments are lacking. The authors believe implementing an opensource intrusion detection framework where all available opensource security frameworks are deployed down at the network interface card, it will go a long way in speeding up development as well as facilitating widespread adoption.

III. Experimental work

A. Objectives

The aim of the experiment presented here is to demonstrate that in collapsing a holistic implementation of security at all layers of the OSI model down to the network interface card, it is possible to mitigate day zero network compromises using our model.

B. Evaluation criteria

The indicators used in this experiment are rudimentary. Within the limits of the experiment, the evaluation criteria is based on the ability of the attack vector employed to successfully achieve privilege escalation on the target. Since the system is at present synthesized on commodity PC hardware, throughput and bandwidth tests are not not required at this stage of the work.

C. Framework

This work is based on the premise that in order for a pervasive security solution to scale efficiently, a distributed, per-endpoint implementation is more desirable. Using this concept we synthesize the HSS/SNiC framework using opensource tools. This has the following advantages:

- Allowing the further extensibility of the implementation
- Providing a trusted and tested platform for intrusion detection research
- Accelerating the development process by leveraging on existing GNU/Linux security toolsets.

D. Experimental method

A dated data-driven exploit is chosen for this experiment to simulate a day zero scenario. For this, the IIS file/directory traversal vulnerability [17] is chosen and is run on a Windows 2000 server with a patch-level less than service pack 3 and running IIS 5.0. A GNU/Linux transparent bridge is inserted inline to the web server and appropriate policies are deployed in it. The policies will take into account the direction of traffic flow, known source and destination MAC addresses.

E. Test bed

The HSS/SNiC model is synthesized using a combination of GNU/Linux and FreeBSD systems with their functionality built upon the Netfilter [18], l7-filter [19] and SNORT [20] frameworks. These are setup as transparent bridges as well as DHCP, DNS, and default gateway devices. The physical topology of the test network is shown in figure 1.

The firewall runs a stable version of FreeBSD with the OpenBSD packet filtering engine [21]. In order to make the network homogeneous, the firewalls also serves as a DHCP and DNS server for the client machines as well as a default gateway. This way, a endpoint device only needs to talk to the default gateway at any point in time.

In reality the attacker and the victim/client is put into separate networks via VLANs defined on a cisco switch to simulate a heterogeneous network and a trunk port is connected to the router/firewall which has the appropriate sub-interfaces and routing defined. An additional host running a tftp server is put in another vlan to simulate a scenario whereby an attacker attempts to make a connection to an external server hosting some exploit code.

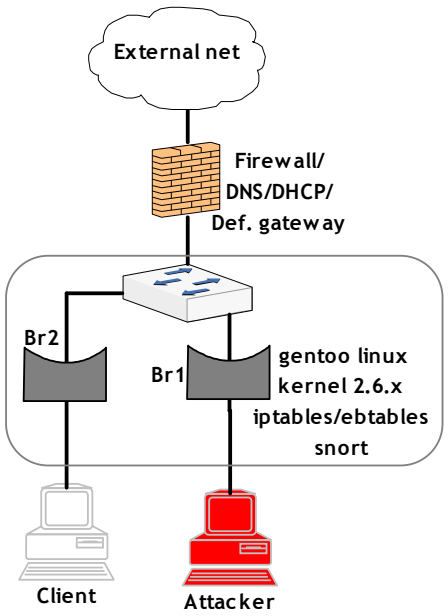

Fig. 1. Physical experimental network.

F. Test procedure

The IIS file/directory traversal vulnerability exploit is broken down into three sessions for the purposes of analysis.

- Connection to port 80 of webserver by http request with malformed URL

  ```
  http://[victim-ip]/scripts/..%255cwinnt/
  system32/cmd.exe?/c+dir+c:\
  ```

- Tftp egress to download netcat binary by malformed http request

  ```
  http://[victim-ip]/scripts/..%255cwinnt/
  system32/cmd.exe?/tftp+-i+rogue_ip+get+
  nc.exe+-l+10000
  ```

- Execution of netcat binary on compromised IIS server to yield command line access

  ```
  nc <victim_ip_> -p 10000
  ```

To simulate the SNiC functionality, a set of iptables/ebtables/arptables rules are deployed on the bridge in-line to the web server as shown in figure 2.

```
#Initialize Netfilter framework to sane state
iptables -F
iptables -X
#Set default FORWARD chain policy
iptables -P FORWARD DROP
#Make and init special chain called x_STATE
iptables -N x_STATE
iptables -F x_STATE
#Drop illegal states
iptables -A x_STATE -m state --state INVALID -j DROP
iptables -A x_STATE -m state --state RELATED,ESTABLISHED -j ACCEPT
#Set up inbound access to IIS web server
iptables -A FORWARD -p tcp -d 192.168.160.33 --dport 80 -j ACCEPT
#drop unwanted external HTTP requests
iptables -A FORWARD -p tcp --dport 80 -m limit --limit 5/minute -j LOG
```

Fig. 2. Security policies applied in-line to IIS web server

It is noted that the rules above are specific to the webserver. The ethos of the framework being proposed is that individual devices can possess varied security credentials based on policy.

In order to verify the integrity of tests, the following precautions were taken

1) The network is monitored in an idle state to ensure that background traffic on the network is not so significant as to influence the results. The bridging protocol traffic is negligent and can be ignored or filtered out.
2) Capture scripts are written to simultaneously capture traffic on both the ingress and egress interfaces of the in-line transparent bridge. This, besides the fact that it gives two capture files to compare with, also provides a means of calculating the forwarding delay between the in/out interfaces.
3) The tests are run multiple times to ensure consistency in the results obtained bearing in mind the deterministic nature of Ethernet traffic.

IV. Results

A. Phase 1

It is observed that phase 1 of the attack succeeds and yields a directory listing of files in the IIS servers root directory as indicated by the characters `'c+dir+c:\'` at the end of the malformed URL request. This also means that any code in the compromised servers *$PATH* variable can be executed.

B. Phase 2

Phase 2 of the attack involves a tftp egress to download a netcat binary. The remote command execution fails and logs on the bridge and tftp server also confirm that no file transfer occurred during this phase of the attack. Table I shows two frames captured from phase 2. The first frame, a tftp request from the IIS server, as initiated by the attacker(via malformed URL) to the tftp server, is repeated 5 times before timing out as shown in the second frame. Spurious Ethernet bridging protocol traffic have been omitted for brevity.

TABLE I
ATTACK PHASE 2: PACKET CAPTURE SHOWING TFTP REQUEST FAILURE AS A RESULT OF MITIGATION

Frame#	Time(s)	Source IP	Destination IP	Protocol	Info
7910	9766.100532	192.168.160.33	192.168.101.252	tftp	Request,file:nc.exe Transfer type: octet
7912	9769.303839	192.168.160.33	192.168.101.252	tftp	Error Code, Code: Not defined, Message: timeout on receive

C. Phase 3

The third phase of the exploit fails since the security policy that mitigates phase 2 of the attack also prevents further privilege escalation via tftp egress.

V. Analysis

Phase 1 of the exploit succeeds as expected since at this stage the synthesized HSS/SNiC doesn't yet possess any layer-7 intelligence so as to determine malicious intent by URL handling, string matching etc.. Hence, the malformed URL request to the server is passed through all security checks and the attack penetrates security looking at the vulnerability in the http service in the server. Phase 2 remote command execution fails as expected since the security policy deployed in the bridge in-line to the server do not explicitly permit egress traffic to an unknown destination.

VI. Conclusions

The experiment presented in this work support the need for a more pervasive and decentralized endpoint security solution and proposes the HSS/SNiC framework as a viable and practical solution. Though present in the original design of the framework, the tests presented here deliberately do not feature any layer-7 intelligence. This is because a day zero scenario is being simulated with no prior knowledge of its nature. It is however trivial to include a single line that introduces the Snort IDS into the iptables execution path hence enabling it detect both data-driven attacks and viruses. This shows that a structured security policy that spans all seven layers of the OSI model can be deployed down to a network interface card hence distributing the burden of endpoint security among endpoint machines while under the strict auspices of a policy-enforcing switch. Recent advances facilitate the deployment of complex IDS signatures in ASICS and network processor units (NPU) with speeds approaching 5Gbps [3] , and more recently 12Gbps [22]. This also makes the proposed solution feasible for use in end-user, SOHO and high bandwidth applications. Further work will also concentrate on the mechanism by which the SNiC authenticate and exchange updates with the HSS.

References

[1] I. Dubrawsky, "Firewall evolution-deep packet inspection," *SecurityFocus, July*, vol. 29, 2003.

[2] H. Song, S. Dharmapurikar, J. Turner, and J. Lockwood, "Fast hash table lookup using extended bloom filter: an aid to network processing," *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 181–192, 2005.

[3] H. Song, T. Sproull, M. Attig, and J. Lockwood, "SNORT offloader: a reconfigurable hardware nids filter," *Field Programmable Logic and Applications, 2005. International Conference on*, pp. 493–498, 2005.

[4] J.-S. E. K. Sung, "A multi-gigabit rate deep packet inspection algorithm using tcam," *GLOBECOM - IEEE Global Telecommunications Conference, GLOBECOM'05: IEEE Global Telecommunications Conference,*, vol. 1, pp. 453–457, 2005.

[5] S. D. S. Kumar, "Algorithms to accelerate multiple regular expressions matching for deep packet inspection," *ACM SIGCOMM Computer Communication Review , Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '06*, vol. 36, Aug. 2006.

[6] L. Tan and T. Sherwood, "A high throughput string matching architecture for intrusion detection and prevention." IEEE Computer Society, 2005, pp. 112–122.

[7] J. Frahim, O. Santos, and D. White, *Cisco Network Admission Control, Volume II: NAC Deployment and Troubleshooting (Networking Technology)*. Cisco Press, 2006.

[8] INFOBLOX, "Building a solid network access control foundation with the infoblox id aware? dhcp solution." [Online]. Available: http://www.webtorials.com/main/resource/papers/infoblox/paper4.htm

[9] Microsoft Corporation, "Introduction to Network Access Protection," 2004. [Online]. Available: http://www.microsoft.com/technet/network/nap/napoverview.mspx

[10] Enterasys Networks, "Trusted end-system solution." [Online]. Available: http://secure.enterasys.com/solutions/secure-networks/trusted_end_system/

[11] R. Glide and X. Shen, "Network appliance for securely quarantining a node on a network," 2006.

[12] Symantec Corporation, "Symantec and Intel collaborate to change security computing model." [Online]. Available: http://www.symantec.com/en/au/about/news/release/article.jsp?prid=20060426_11

[13] W. de Bruijn, A. Slowinska, K. van Reeuwijk, T. Hruby, L. Xu, and H. Bos, "Safecard: a gigabit ips on the network card," *Proc. of 9th International Symposium on Recent Advances in Intrusion Detection (RAID'06), Hamburg, Germany, September*, 2006.

[14] T. Garfinkel and M. Rosenblum, "A virtual machine introspection based architecture for intrusion detection," *Proceedings of the 2003 Network and Distributed System Security Symposium (NDSS)*, 2003.

[15] H. Bos and K. Huang, "Towards software-based signature detection for intrusion prevention on the network card," *Proc of the 8th International Symposium on Recent Advances in Intrusion Detection (RAID)*, 2005.

[16] M. Otey, S. Parthasarathy, A. Ghoting, G. Li, S. Narravula, and D. Panda, "Towards nic-based intrusion detection." Washington, D.C.: ACM, 2003, pp. 723–728.

[17] Compter Emergency Response Team, "Cert advisory ca-2001-12 superfluous decoding vulnerability in iis." [Online]. Available: http://www.cert.org/advisories/CA-2001-12.html

[18] P. Russell, *Netfilter: Firewalling, NAT and packet mangling for Linux 2.4*. Netfilter.org, 2001. [Online]. Available: Available: http://www.netfilter.org/documentation/index.html

[19] J. Levandoski, E. Sommer, and M. Strait, "Application layer packet classifier for linux." [Online]. Available: http://l7-filter.sourceforge.net/

[20] B. Caswell and M. Roesch, "SNORT: The open source network intrusion detection system," 2004. [Online]. Available: http://www.snort.org

[21] D. Hartmeier, "Design and performance of the OpenBSD stateful packet filter (pf)," *Proceedings of the USENIX Annual Technical Conference, Freenix Track*, pp. 171–180, 2002.

[22] Checkpoint Software Technologies Ltd., "Checkpoint VPN-1 power sets new industry standard for security performance." [Online]. Available: http://www.checkpoint.com/press/2007/r65vpn-1power.html